Predicting Stock Market Prices using Deep Learning by Tensor Flow

Yajnavalkya Bandyopadhyay[1], Sandip Roy [2], and Siddhartha Chatterjee [3]

[1] Department of Civil Engineering
Techno India College of Technology, Kolkata, WB 700156, India
Email : yajnab@gmail.com
[2] Department of Computer Science and Engineering, Asansol Engineering College, Asansol, WB 713305, India
Email : sandiproy9500@gmail.com
[3] Department of Computer Application, DSMS Group of Institutions
Durgapur, WB 713206, India
Email : siddhartha.chatterjee31@gmail.com

Abstract. Stock Market investors are in need of good quality stock prediction system to maximize their profit and minimize their loss, which makes the reason for prediction system to be in action.The Prediction system uses a deep learning based neural network system for prediction. TensorFlow(A neural network from Google) is used for designing the Deep Learning network where data of April to July values are provided for training and the August values are predicted and compared with the old values of that month. Deep Learning offers one of the best ways to predict stock market values using historical Data.

Keywords: ANN, Tensorflow, Deep learning, Yahoo Finance, Prediction

1 Introduction

Stock Market values are very difficult to predict using statistical models. The prices if considered for short time they change due to News or Influencing media, for long term they vary because of firm conditions and market demands and supply. However deep learning Neural Network model can give efficient stock market predictive values from historical data [4][5].
In our work we have taken values from April to August with difference of 1 minute. The Neural Network having sigmoid function with backpropagation support. Leading 80% of the data was used for training the neural network and compared to the rest 20% of the data.

The work is done on Python 3.5.3 using Google TensorFlow Library for designing the Neural Network. There has been growing interest in whether deep learning can be effectively applied to problems in finance, but the literature (at least in the public domain) still remains limited. With the increasing availability of high-frequency trading data and the less-than-satisfactory performance of

Electronic copy available at: https://ssrn.com/abstract=3586407

existing models,comprehensive studies that objectively examine the suitability of deep learning to stock market prediction and analysis seem opportune. The ability to extract abstract features from data, and to identify hidden non-linear relationships without relying on econometric assumptions and human expertise, makes deep learning particularly attractive as an alternative to existing models and approaches.

1.1 Technical preliminaries : TensorFlow and Artificial Neural Network

TensorFlow is a great piece of software and currently the leading deep learning and neural network computation framework. It is based on a C++ low level backend but is usually controlled via Python. TensorFlow operates on a graph representation of the underlying computational task. This approach allows the user to specify mathematical operations as elements in a graph of data, variables and operators. Since neural networks are actually graphs of data and mathematical operations, TensorFlow is just perfect for neural networks and deep learning.

Artificial Neural Networks (ANNs) require a careful selection of the input variables and network parameters such as the learning rate, number of hidden layers, and number of nodes in each layer in order to achieve satisfactory results [1]. It is also important to reduce dimensionality to improve learning efficiency. On the other hand, deep learning automatically extracts features from data and requires minimal human intervention during feature selection. Therefore, our approach does not require expertise on predictors such as macroeconomic variables and enables us to use a large set of raw-level data as input. Ignoring the factors that are known to predict the returns, our approach may not be able to outperform existing models based on carefully chosen predictors. However, considering the fast growth of deep learning algorithms, we believe this research will serve as a milestone for the future research in this direction. [2] also predict that deep learning will play a key role in financial time series forecasting.

1.2 Related Works

Stock Market prediction has been a trivial work for the Scientists and Mathematicians. At early stage Mathematics and statistics could predict the Stock Prices at lower efficiency using Regression models [11][12]. With the change of time Artificial Neural Networks [9][10]using historic values took its place. Deep Learning Algorithms has proven to be best in this field as we can not only train the values using historical data but also using datas like Media, News , demand and supply.

1.3 Our Contribution

The fundamental contribution of the paper is as follows:

Electronic copy available at: https://ssrn.com/abstract=3586407

1. we have used Python 3.5.3 with TensorFlow 1.5.3 giving Deep Learning aspect for prediction of the future values using historical values.
2. Multiple(4 in number) hidden layers are present in the system which makes the learning rate to be higher with better training support.
3. With an interval of one minute, series of data values are collected from consecutive four months record. The Neural Network having sigmoid function with backpropagation support. Leading 80% of the data was used for training the neural network and compared to the rest 20% of the data.

2 Deep Learning for Stock Market Prediction

2.1 Testing and Training Data

The system shifts the each stock values by 1 minute to make the training and testing dataset exclusive so that we do not predict the current data. The testing and training data are converted to different .CSV files for discriminating testing and training values.
The time series bootstrap re-sampling is used which involves repeated samples from the remainder of the seasonal decomposition of the time series in order to simulate samples that follow the same seasonal pattern as the original time series but are not exact copies of its values.

2.2 Data Scaling

Most neural network architectures benefit from scaling the inputs because most common activation functions of the networks neurons such as tanh or sigmoid are defined on the [-1, 1] or [0, 1] interval respectively. Nowadays, rectified linear unit (ReLU) activations are commonly used activations which are unbounded on the axis of possible activation values. However, sklearns MinMaxScaler Library Package is used in this case to easily scale the data in Python. The testing and training data are scaled differently to avoid mutually inclusive scaling which can cause errors in prediction.

2.3 Optimizer

The optimizer is used for necessary computations for network weight and bias variables during training. This tool indicate the direction of the gradient which indicate the direction of the weights and biases have to be changed to minimize the network cost function.
Here Adaptive Gradient Algorithm (AdaGrad) which is a stochastic algorithm, which is used that maintains a per-parameter learning rate that improves performance on problems with sparse gradients.

Electronic copy available at: https://ssrn.com/abstract=3586407

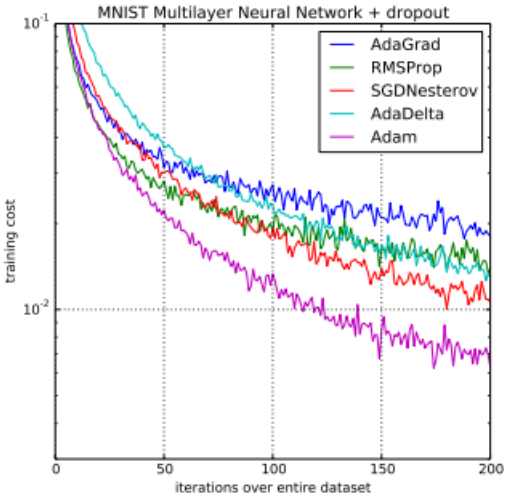


Fig. 1. Training of multilayer neural networks using dropout stochastic regularization

2.4 Placeholders and Variables

Tensorflow requires Placeholders for fitting the Input and Outputs. One Placeholder gets the input in time=t and another placeholder gets the output at time = t+1. Variables are another for requisite Tensorflow. They form the flexible changing data storage during the execution of the program. Weightage and Bias are the changing values which are stored in the variables.

Designing the Network The Neural network is designed keeping the model having 4 hidden layer where the number of Neurons are decreased by halving the Neuron numbers from 1024,512,256 and 128 respectively in respective hidden layer to compress the data and finally one output neuron as Placeholder for output.

2.5 Cost Function

The cost function of the network is used to generate a measure of deviation between the networks predictions and the actual observed training targets. For regression problems, the mean squared error (MSE) function is commonly used. MSE computes the average squared deviation between predictions and targets. Basically, any differentiable function can be implemented in order to compute a deviation measure between predictions and targets. The MSE function in mathematical equatio obeys as follow:

$$E = \frac{1}{m}\sum_{n=1}^{m}(h(x) - y_i)^2 \tag{1}$$

Electronic copy available at: https://ssrn.com/abstract=3586407

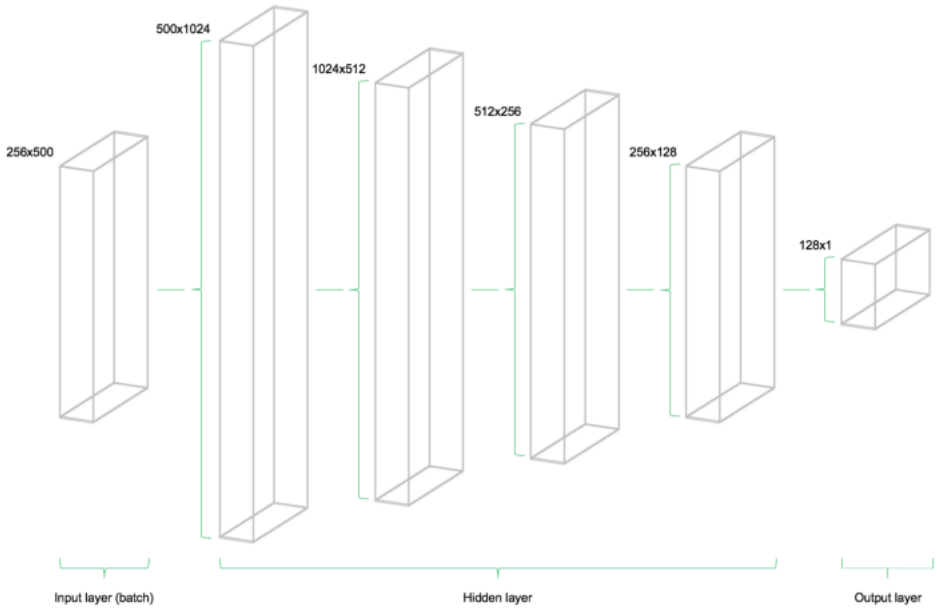


Fig. 2. Neural Network Model

where h(x) stands for mean value of the data and y_i stands for the i^{th} value.

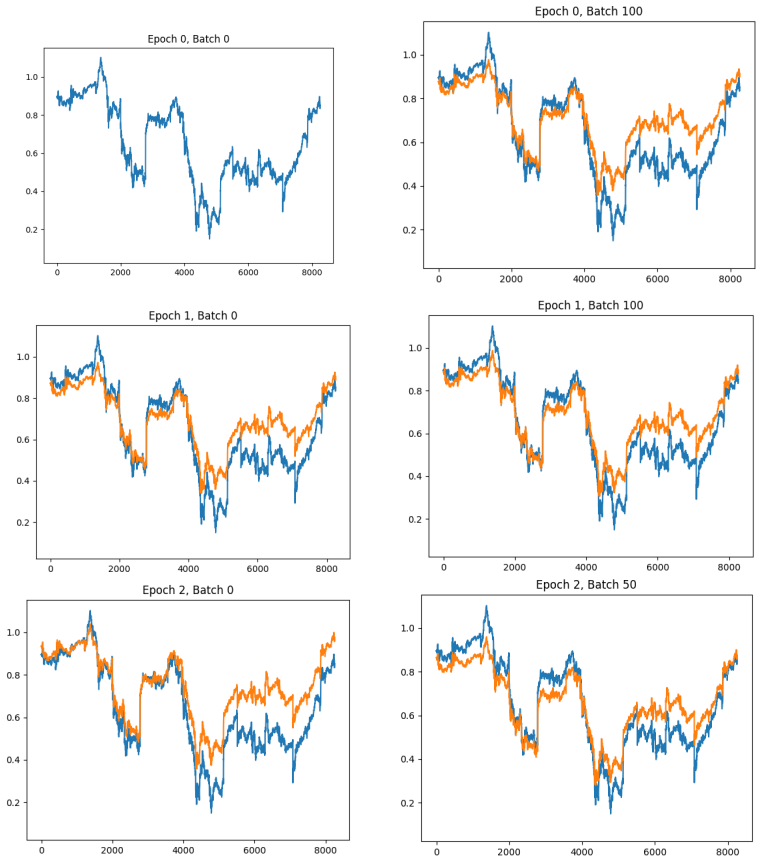
2.6 Optimizer

After having defined the placeholders, variables, initializers, cost functions and optimizers of the network, the model needs to be trained. Usually, this is done by minibatch training. During minibatch training random data samples of $n = batch_size$ are drawn from the training data and fed into the network. The training dataset gets divided into $\frac{n}{batch_size}$ batches that are sequentially fed into the network. At this point the placeholders X and Y come into play. They store the input and target data and present them to the network as inputs and targets.
A sampled data batch of X flows through the network until it reaches the output layer. There, TensorFlow compares the models predictions against the actual observed targets Y in the current batch. Afterwards, TensorFlow conducts an optimization step and updates the networks parameters, corresponding to the selected learning scheme. After having updated the weights and biases, the next batch is sampled and the process repeats itself. The procedure continues until all batches have been presented to the network. One full sweep over all batches is called an epoch. The training of the network stops once the maximum number of epochs is reached or another stopping criterion defined by the user applies.

Electronic copy available at: https://ssrn.com/abstract=3586407

3 Results

The Deep Learning model was trained using 80% of the feeded data. Rest 20% was predicted using the model and compared with the original data. On performing that we get the following result where blue coloured graph indicates the actual data and yellow colour indicates the predicted data. The absicca and the ordinate in the graphs shows time variant and Price respectively. A full epoch is completing the training for 100 batches where the model runs to solve the model of the graph. After a few epoch we get the model prediction comes close to the original data. Upon running with the provided settings these training results in each epoch and batches we get,



Electronic copy available at: https://ssrn.com/abstract=3586407

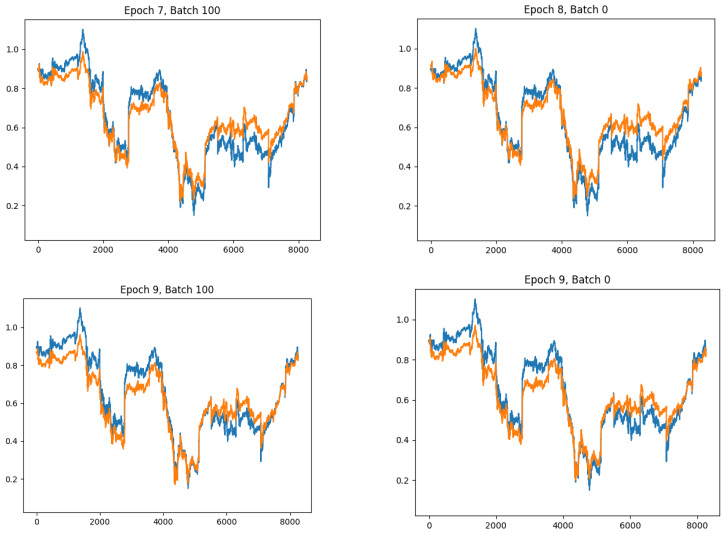


Fig. 3. Resulting Graphs. Yellow stands for prediction values and blue stands for original training values

4 Conclusion and Futute scope of work

The paper deals with single type of input which is only the historical data and TensorFlow maps the function to get the prediction values on. The model can be made more robust using real life influencing media such as news which can make the prediction more efficient. No doubt this model acts better than statistical data models which fails on the learning and backprogation support. This model can be used for predicting of gold, silver and currency exchange and other type of data. Analysing big data using hadoop and concurrent live data for prediction can be a big goal to accomplish.

References

1. Hussain et al, Neural networks for financial time series prediction, 2008
2. Konstandinos Chourmouziadis, Prodromos D. Chatzoglou, An intelligent short term stock trading fuzzy system for assisting investors in portfolio management, Expert Systems With Applications, 2016
3. M. R. Vargas, B. S. L. P. de Lima and A. G. Evsukoff, "Deep learning for stock market prediction from financial news articles," 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), Annecy, 2017, pp. 60-65. doi: 10.1109/CIVEMSA.2017.7995302
4. Y. Zhai, A. Hsu, S. Halgamuge, "Combining news and technical indicators in Daily stock price trends prediction", Advances in Neural Networks, pp. 1087-1096, 2007

Electronic copy available at: https://ssrn.com/abstract=3586407

5. 2. J. Bollen, H. Mao, X. Zeng, "Twitter mood predicts the stock market", Journal of Computational Science, vol. 2, no. 1, pp. 1-8, 2011.
6. Singh, R. and Srivastava, S. Multimed Tools Appl (2017) 76: 18569. https://doi.org/10.1007/s11042-016-4159-7
7. Eunsuk Chong Chul wooHa Frank C.Park, "Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies", Expert Systems with Applications, 2017
8. R. Akita, A. Yoshihara, T. Matsubara and K. Uehara, "Deep learning for stock prediction using numerical and textual information," 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, 2016, pp. 1-6.
9. http://www.euclidean.com/the-intelligent-investor
10. Sabaithip Boonpeng, Piyasak Jeatrakul, "Enhance the performance of neural networks for stock market prediction: An analytical study", Digital Information Management (ICDIM) 2014 Ninth International Conference on, pp. 1-6, 2014.
11. Girija V Attigeri, Manohara Pai M M, Radhika M Pai, Aparna Nayak, "Stock market prediction: A big data approach", TENCON 2015 - 2015 IEEE Region 10 Conference, pp. 1-5, 2015, ISSN 2159-3450.
12. H. L. Siew and M. J. Nordin, "Regression techniques for the prediction of stock price trend," 2012 International Conference on Statistics in Science, Business and Engineering (ICSSBE), Langkawi, 2012, pp. 1-5. doi: 10.1109/ICSSBE.2012.6396535

Electronic copy available at: https://ssrn.com/abstract=3586407